H－01
中文教育电脑使用手册

普 乐 电 器 公 司

八七年一月

H－01

中文教育电脑使用手册

普　乐　电　器　公　司

八七年一月

前　言

近年来为了加快开展我国的计算机教学，推广普及计算机的应用，先后从国外引进了多种灵巧型微机，如 LASER 310，PC8300，COMX35 等等。它们在这段时期内的计算机教学实践中，发挥了很好的作用，成为提高学生智力水平的有效工具，是中小学生的良师益友，被人们亲切地称为“娃娃机”。但是，由于它们的固有的局限性，主要是不能方便地处理汉字，使它们难于在计算机辅助教学这一蓬勃发展的广阔领域内一显身手。

因此，我公司结合我国的应用特点分析了一些国外进口的机器，经过精心设计，生产出了适合中国应用特点的H—01汉字电脑，解决了教学机的汉字处理问题。由于采用了汉字压缩新技术，使得小小的H—01机具有处理3755个汉字的能力；向用户提供具有32K用户空间的良好运行环境。实现了BASICⅡ（除磁盘部份）的全部功能。其图形功能和电子琴功能更有独到之处。使之成为中小学教学的不可多得的理想工具。不仅如此象在部队系统，大专院校、工矿企业等等各个行业的计算机教学，工业过程控制，专用仪器设备等应用项目中，也是H—01大显身手的好地方。

欢迎您使用H—01中文电脑，它将给您的学习、工作和生活带来极大的帮助和乐趣。

但是，由于时间，水平有限，本手册中难免有错误之处，请教正。且应注意的是，本手册并不是一本完整的教科书。有关BASIC程序设计方面的技术，建议参阅有关书籍。

普乐电器公司

目　　录

第一章　概　　要 …………………………………… (1)

§1.1　部件简介 …………………………………… (1)

1.　显示器和键盘 …………………………………… (1)

2.　外存贮器 …………………………………… (2)

3.　打印机 …………………………………… (3)

§1.2　如何启动 H—01 机 …………………………………… (4)

1.　接通电源 …………………………………… (4)

2.　调整显示器 …………………………………… (4)

3.　进入 H-BASIC …………………………………… (5)

4.　键盘及其特殊功能键 …………………………………… (6)

第二章　H-01 机的基本操作 …………………………………… (10)

§2.1　如何输入汉字 …………………………………… (10)

§2.2　显示方式 …………………………………… (11)

1.　中西文兼容方式 …………………………………… (11)

2.　全字符方式 …………………………………… (11)

§2.3　H—01 机的作图 …………………………………… (11)

§2.4　光标控制 …………………………………… (12)

§2.5　第二字符集 …………………………………… (12)

§2.6　电子琴演奏 …………………………………… (13)

§2.7　打印方式控制 …………………………………… (15)

§2.8　造字 …………………………………… (16)

§2.9　磁带操作 …………………………………… (18)

第三章 运行程序的基本过程……………… (22)
第四章 命令、编辑状态 ………………………… (25)
§4.1 命令状态 …………………………………… (25)
§4.2 编辑状态……………………………………… (30)
第五章 H-BASIC 语言……………………… (34)
§5.1 基本成分……………………………………… (34)
§5.2 输入/输出语句……………………………… (38)
§5.3 程序语句 …………………………………… (46)
§5.4 字符串………………………………………… (54)
§5.5 数学函数……………………………………… (60)
§5.6 特殊功能和逻辑运算……………………… (62)
附录1. H-BASIC保留词汇 ………………… (70)
附录2. 程序限制(数据范围)……………… (71)
附录3. 常用内存数量…………………………… (71)
附录4. 动态(运行时)内存分配 ………… (72)
附录5. 错误代码 ……………………………… (73)
附录6. ASCII 代码和图示代码 ………… (76)
附录7. 外引脚定义……………………………… (77)
附录8. 监控主要子程序调用……………… (81)
附录9. 可用I/O口 …………………………… (83)
附录10. 特殊功能键及键盘命令表……… (84)

第一章 概　要

图1-1所示为典型的H—01计算机系统。

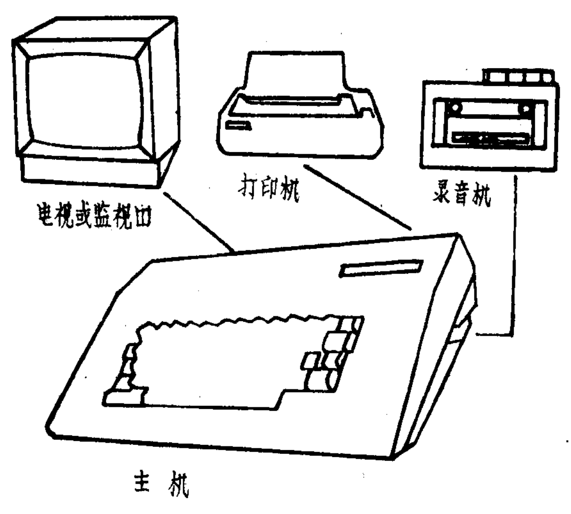


图1—1　H—01机的基本系统

它包括H—01主机、显示器、外存贮器和打印机等几个部件。下面我们分别做介绍。

§1.1. 部件简介

1. 显示器和键盘

人们通过键盘和显示器与H—01机联系。通过键盘告诉

计算机做什么，通过显示器把执行结果显示出来。

显示器可以采用普通的家用彩色或黑白电视机。但由于H—01机不支持彩色功能，所以用黑白电视机和彩色电视机显示效果相同。注意，老式的黑白电视机往往沒有75Ω天线插孔，这需要打开电视机机盖，将连线直接连接在高频头上。

在计算机系统中，监视器产生的图象比电视机清晰，稳定，但监示器不能用来收看电视节目。图1-2示出H—01机和电视机及监示器的连接方式。

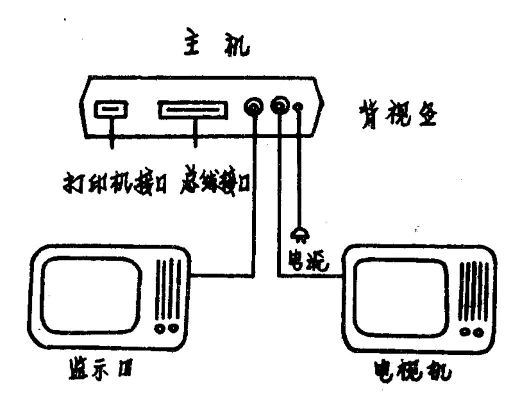


图1-2 H—01机和电视机及监示器的连接

2. 外存贮器

放在H—01机内的程序不能长期保存，关机后便会消失。所以需要把机内的程序保存下来。H—01机采用普通的家用录音机作为外存。图1-3示出 H—01 机和录音机的连接方式。用家用录音机作为计算机外存贮器的要求比存放音乐信号的要求要高得多，你应该选用质量好的磁带如 SONY、

TDK等。你的录音机也应经常清洗，作消磁处理。一般来说，在用你的录音机播放音乐磁带时，没有失真和背景噪音、声音清晰悦耳便可以使用了。关于录音机的使用注意事项参见第二章第九节。

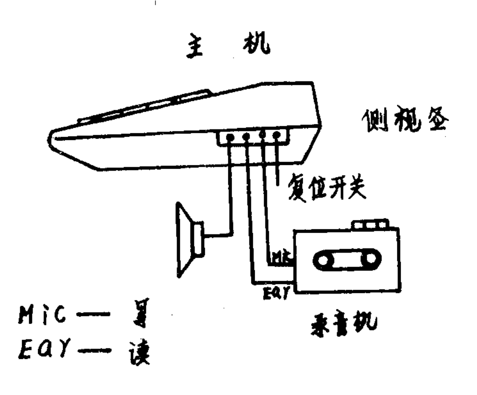


图1-3 H—01机和录音机的连接

3. 打印机

打印机通过一个专用打印机接口与H—01机相连。图1-4示出连接方式。

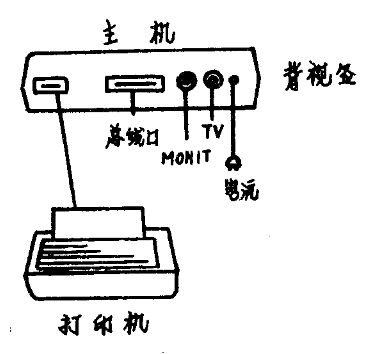


图1-4 H—01机与打印机的连接

打印机的种类很多。H—01机能够支持和MX-80兼容的9针打印机。每台打印机上都有一些控制开关，其中有一个是自动换行控制开关，如果你发现打印机打印时纸不向前走，请参阅打印机的使用说明书，将自动换行开关置于ON位置。

§1.2 如何启动H-01机

1. 接通电源

首先你必须按前节的介绍把整个系统连接好，检查正确无误。然后，打开电视机（或监示器），把音量关至最小并置于2频道（若你用电视机作显示）。

电源开关位于H—01机左侧，接通电源开关，数秒钟后，便可听到由H—01机内发出“吱”的一声。这说明，H—01机已经准备就绪，此时，键盘上的电源指示灯应发亮。

若沒有听到声音电源指示灯未亮。应将电源线拔掉，仔细检查电源连线是否牢靠，保险丝是否完好，然后重新上电，若指示灯发亮，沒有声音，则必须将电源关掉，这一般是机器內部有损坏。

在加电时，用戶必须注意：当机器断电后，必须等待5～10秒后方可重新上电，否则H-01机无法进入正常工作。

2. 调整显示器

一旦听到H-01机发出‘吱’的声音，这意味着一切正常，屏幕上应出现图1—5所示画面。这时，应仔细调整频道微调，亮度和对比度，使画面稳定、清晰，有时还需要调整水平同步或垂直同步。如果你使用的是监示器，调整工作要简

H－01型中文教育电脑
普乐电器公司

造字起始地址47104　　　英

图1—5　H—01机开机后的画面

单的多，若是画面跳动不止，调垂直同步无效，将监视器后面的阻抗变换开关换一个位置试试。

3.进入 H-BASIC

当你调整好显示画面后，H-01 机准备好接受你的指令。现在你可暂不关心画面上提示的造字起始地址。随便按一个键（除几个特殊功能键），H-01机会立即转入第二个画面，提示‘内存最高地址？’，我们在后面介绍这个提示的含义。现在，你只要按 RETURN 键 H-01机便会转入 BASIC。这时的屏幕显示如图 1—6。

```
内存最高地址?
H-BASIC
准备好
>
```

图1—6　H-01机进入BASIC后的画面

4. 键盘及其特殊功能键

H-01机的键盘如图1—7所示。在键盘上打入任何字

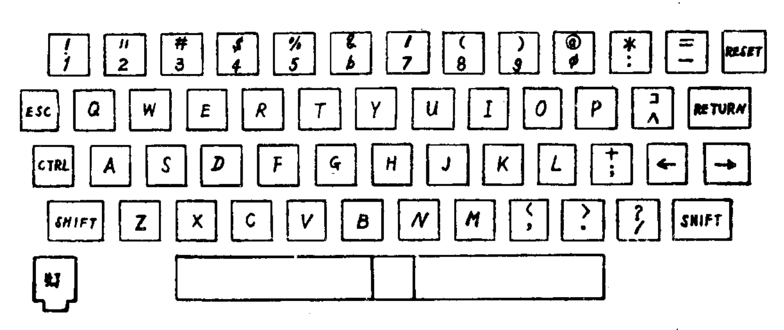


图1—7 H-01机的键盘

符都不会造成H-01机的损坏。

H-01机修改了键盘上几个键的含义，这主要是为了方便用戶使用或满足H-01机某些特殊功能的需要。下面分别介绍各个功能键。

1) 复位键（RESET键）

复位键是一个很特殊的键，按下该键，计算机将从头开始初始化工作，其效果与重新上电相同，但要安全可靠得多。主要用于使机器从某种锁死状态退出。如调磁带失败、发了打印命令而有沒连接打印机等。

由于RESET键非常靠近RETURN键，为了防止误操作造成內存程序的丢失，H-01机已将复位键移到机壳侧面，而将原RESET键重新定义为下移光标键（↓键）。

2）回车键（RETURN键）

在键盘上不断打入字符，屏幕上就会不断地将这些字符显示出来并存入内存，直到按下回车（RETURN）键，机器才翻译是打入了什么指令。如果打入的指令机器能够明白它就会产生相应的动作，否则，在屏幕上显示相应的错误信息。

3）上字键（SHIFT键）

除了个别键，H-01机上的每个键都有上、下档之分，对于控制键，上、下档实现不同的功能控制，对于字符键，上、下档产生不同的字符。我们用SHIFT—表示按住SHIFT不放，同时按下另一个键。例如：SHIFT-3 产生的字符是#、SHIFT-A 产生的字符是 a。

4）英汉转换键

该键位于键盘左下角，用来转换系统输入状态。在H-01机的中西文兼容方式下，按动该键可以看到屏幕右下角的系统提示在变化。当系统提示为'英'时，键入内容作为ASC字符处理。当提示为'汉'时，允件用拼音或区位码输入汉字。关于汉字输入方法参见§2.1。

在状态'英'下，输入SHIFT-英/汉键，装有五笔划汉字输入模块的机器会在系统状态提示处显示出一个'五'，表示机器此时允许用五笔划方式输入汉字，对沒装该模块的机器将不产生影响。

5）进入电子琴

按下 SHIFT-CTRL 键可进入键盘电子琴。关于电子琴演奏参见 § 2.6。

CTRL 键的代码是10H，在运行 FORTH 程序时，该键是打印机联机开关，在BASIC中为非法字符。

6）中断键（BREAK键）

中断键定义在 ESC 键上。按下该键就可停止正在执行的动作，并返回命令状态。

7）幂键（]键）

幂键定义在^键上，用于进行指数运算。

例10 PRINT X^2；表示要打出 X^2值。

（注：屏幕显示的幂符号是“[”。）

8）退格键（BS键）

退格键定义在 ← 键上，其功能是光标退格，并删除刚打入的字符。

9）屏幕硬拷贝键

屏幕硬拷贝键定义在SHIFT—→键上。按下该键即可把显示屏上的内容全部用打印机拷贝下来。但未连接打印机时，会将机器锁死。

10）暂停键

暂停键定义在SHIFT—RESET上，该键的代码为60H显

示ⓐ。按下该键便可暂时停止程序的执行，再按任何键程序便继续执行。

11）字符切换键

字符切换键定义在SHIF-←键上。主要用于造字，按下该键便可把所造的字从键盘上一一调出，再按该键又返回ASCⅡ键盘。在H-01机上的其它键都是使用者熟悉的，它们是26个英文字母，从0～9十个数字，以及一些标准的符号键。

H-01机的任何一个键都具有自动重复功能。

第二章　H—01机的基本操作

§2.1　如何输入汉字

按一下'英/汉'转换键，使屏幕右下角的系统提示为'汉'，然后键入汉字的汉语拼音。拼音显示在屏幕左下角，输入完一个字的拼音后，按空格键，如果拼音正确屏幕下方立即出现十个汉字，否则显示'输入错'。若这十个汉字中沒有你所需要的汉字，再按一次空格键，又出现另外十个汉字，依次类推，直到出现你所需要的汉字为止。十个字对应于键盘上（1－10）的数字键（10用0代替），用数字键选出你所需要的字。

如果你按了不止一次空格键，显示的汉字中已有读音和你所输入的拼音不同的汉字，这说明你所需要的字不在国家标准一级汉字中。按'一'键，可回到前次提示的十个汉字。例如：要输入'欢迎'二字，在汉字状态下输入'HUAN'按空格键后，系统行提示：

欢环桓还缓　换患唤痪豢

键入'1'，即在屏幕光标处出现'欢'字。再键入'YING'按空格键，屏幕系统行提示：

英樱婴鹰应　缨莹萤营荧

这十个字中沒有迎字，再按空格，又有十个汉字出现：

按‘2’，迎字紧接着欢字出现在屏幕上。

用区位码键入汉字时，系统行只提示已输入的区位码，例如，当你键入‘1601’后，‘啊’字直接出现在光标处。

用户必须注意的是：在 H-BASIC 中，字符串常数一定要用引号引起来，无论是字符或汉字都是如此，否则将产生错误。

§2.2　显示方式

1．中西文兼容方式

这种方式也可叫做方式 0。用命令（或语句）MODE(0)进入。开机后直接进入该方式。在该方式下，每屏幕显示12行×42个字符（或21个汉字），其中第12行为系统提示行（用户不能使用）。这种方式支持汉字的输入及输出。

2．全字符方式

这种方式也叫做方式 1，用 MODE（1）进入。在该方式下，每屏幕显示16行×42个字符。在这种方式下，无论是系统提示的汉字，还是用户程序输出的汉字，均显示空格，并且也不响应英/汉转换键。

在用 MODE 进行方式转换时，H-01机不清屏，转换前屏幕上的内容转换后仍然保留。

§2.3　H-01机的作图

H-01机支持高分辨率作图。作图一般在方式 1，否则Y方向会有间隔。但方式 1 不支持汉字，有两种方法可以达到图形汉字兼顾。

1. 利用方式转换不清屏的特点，先将汉字显示在屏幕上，然后进入方式 1 作图，此时机器停留在方式 1。

2. 在方式 0 下运行下面一段程序即可消去行间距，注意运行完该程序后 H-01机仍处于方式 0。

程序：
```
10 OUT 96，4：OUT 100，19：OUT 96,5：
   OUT 100， 4
20 OUT 96，9：OUT 100，15：OUT 96,7：
   OUT 100,16
```

有关作图语句的使用参见 MODE，SET和RESET语句。

§2.4 光 标 控 制

H-01机的光标可用如下语句控制

POKE 16888，32　　　　关光标

POKE 16888，255　　　开光标

向地址16888中置入不同的数值，可以得到几种不同大小的光标。

§2.5 第 二 字 符 集

H-01机的字符集分为两部分（不包括汉字）。第一部分94个为标准 ASCⅡ字符集。你可以在键盘上看到并按动相应键将其输入机器；第二部分为94 个由常用数学符号等组成，按一下 SHIFT-←键，就可以把这些字符由键盘一一调出，再按一下 SHIFT-←键，又返回 ASCⅡ键盘。

这些字符在程序中可以作为字符处理，作为字符常数使用时注意两边需要用引号，另外，也可以用函数 CHR $ 处理它们，其编码值由126—219，详细的字符形状、码值及相

应键各位请参见附录6。

§2.6 电 子 琴 演 奏

按下 SHIF-CTRL 后，H-01机的键盘就变成电子琴的琴键。图2—1为键盘上的音调分布，是按1＝C给出的。演奏完电子琴后，按‘汉/英’键退回。

下面是一首曲子，用程序演奏如下：

I＝E 2/4

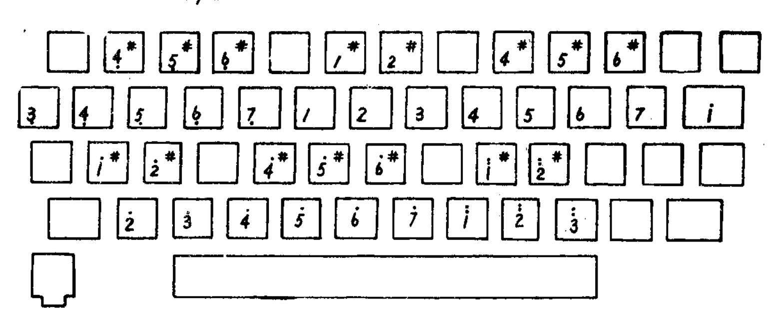


图2—1 电子琴音调分布图

11 55 | 66 5 | 44 33 | 22 1 | 55 4 | 33 2 |

55 44 | 33 2 | 11 55 | 66 5 | 44 33 | 22 1 |

编程演奏如下：

```
10 DATA 21 4，21， 4，28， 4，28， 4，30， 4
20 DATA 30， 4，28， 6，26， 4，26， 4，25， 4
30 DATA 25， 4，23， 4，23， 4，21， 6，28， 4
40 DATA 28， 4，26， 6，25， 4，25， 4
50 DATA 23， 6，28， 4，28， 4，26， 4，26， 4
```

```
60  DATA 25, 4, 25, 4, 23, 6, 21, 4, 21, 4
70  DATA 28, 4, 28, 4, 30, 4, 30, 4, 28, 6
80  DATA 26, 4, 26, 4, 25, 4, 25, 4, 23, 4
90  DATA 23, 4, 21, 6
100 OFR I=1 TO 41
110 READ F, D
120 SOUND F, D
130 NEXT I
140 END
```

每键入一次，除了其它响应外，机器还发出'吱'的一声。这个声音的音高及有无都可由用户重新设定。H-01机共有三个八度音程，共36个音，对应于数字 1～36。频率和音调时间的对应表如下：

频率码	0	1	2	3	4	5	6	7	8	9	10	11……12	36
音 调	无	3 E	4 F	4#	5 G	5#	6 A	6#	7 B	1 C	1#	2 D ……2#	3
时间码		1	2	3	4	5	6	7	8	9	10		
长 度	1/16	1/8	¼	3/8	½	3/4	1	$1\frac{1}{2}$	2	3			

执行：

POKE	16893, 0	取消按键音响。
POKE	16893, 1	发最低音。

⋮

⋮

POKE　16893，36　发最高音。

若置入数大于36，H-01机取消按键消抖延迟，键盘反应极快，这主要适用于游戏或特殊場合。

§2.7　打印方式控制

H-01机提供7种不同的打印方式，用戶可以根据要打印输出的內容选择，以达到又快又省的目的。

1．方式0

进入用　POKE　16860，0

这种方式为系统设定方式。该方式的特点：两行汉字之间不空点，ASC字符间空7点，并且是唯一能够正确执行屏幕硬拷贝的打印方式，适应于汉字不多的输出要求。

2．方式1

进入用　POKE　16860，1

两行ASCⅡ字符间隔3线，汉字间有重线，适用于输出內容中汉字很少的情况。

3．方式2

进入用　POKE　16860，2

两行汉字之间空有3线，ASCⅡ字符间空10线，适用于以汉字为主的输出要求。

4．方式3

进入用　POKE　16860，3

这是唯一一种直接使用打印机字模的方式，速度快，密度高，缺点是无法打印汉字。所有汉字一律换为两个空格。两行字符之间间隔多少取决于进入本方式前的打印方式。

5. 方式4

进入用 POKE 16860，16

与方式 0 相同，但输出字的大小为方式 0 的一半。

6. 方式5

进入用 POKE 16860，17

与方式 1 相同，但输出字的大小为方式 1 的一半。

7. 方式6

进入用 POKE 16860，18

与方式 2 相同，但字的大小为方式 2 的一半。

打印方式和显示方式无关。在显示方式 1 下，而在打印方式 0 下，仍可打印输出汉字。

§2.8 造　字

H-01 机的造字起始地址提示在开机后出现的第一帧画面上，一般应为47104。你造好的字模点阵由这个地址开始逐个送入内存。H-01 机允许在一个8(宽)×16(高)点的点阵中造字。比如，你需要造一个长方形图案，先排出要求的字模如下：

0000　0000　　　00H

0000　0000　　　00H

0111	1110	7EH
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0100	0010	42H
0111	1110	7EH
0000	0000	00H
0000	0000	00H

图中的'0'表示暗点，'1'表示亮点，亮点组成了我们需要的长方形。若将上面的字模看成二进制数的0或1，每行相应的两个十六进制数化成十进制数，（例如 7EH = 7×16 + 14 = 126，42H = 4×16 + 2 = 66等），按从上到下的顺序 用 POKE 语句打入内存即可。这些新造字会复盖掉 H-01 机第二字 符集中的字符。如果你希望将该字符定义在某个键上，起始地址可由下式算出：

首址 = (相应键的ASCⅡ码 − 32) × 16 + 47104

若将前面的长方形定义在 A 键上，A的 ASCⅡ码为65，所以

首址 = (65 − 32) × 16 + 47104 = 47632

将长方形的字模全部内容打入内存后，按动键盘转换键，再按 A 键调出的便是该长方形。

新造字和汉字一样，只能作为字符处理，作为字符常数

出现，不能由CHR $函数处理。第二字符集中的字符仍可由CHR $函数调出。

字模的形状与系统无关，用户可以在8×16点阵里任意排，H-01机均给以处理，这实际上意味着H-01机在处理、中文的同时，还可以处理第三种文字。或者定义一些基作图符号如象电子电路中的电阻，电容等，用编辑文本的办法来编辑这些子图形，可以容易地‘拼’出一张电路图，利用此功能也可以编出适于幼儿园小朋友的‘图形积木’程序。

值得指出的是，在显示方式1时，只由下向上显示12线，新造字若须在两种字模下都能使用，高4线须全为0。

§2.9 磁带操作

许多人都曾为存取磁带文件而大伤脑筋。这里面固然有其它原因，但用户缺乏经验，使用不当，调整方法不对也是常有的，为了帮助你顺利使用磁带，节省宝贵的时间，希望你在使用磁带存取文件之前详细阅读本节内容，不要忽略掉其中任何一条注意事项，这些都是我们及许多用户的经验之谈。

1. 录音机和磁带的选用

H-01机的录音机接口是为了和普通低档家用机配合使用的，某些专门用作计算机外存的录音机反而不太好用。这主要是录音电平不合适。H-01机送给录音机的写带电平是40mV左右，而专用于计算机的外存录音机的录音电平往往在5～10mV左右。如果你是一个有经验的用户，而且已经有了这种录音机，你可将写带电平衰减后送往录音机，否则

不要动手。

为保险起见，必须选用高质量磁带。因为磁带上的任何一点缺陷都会造成读带失败或漏读、错读、走带机构不灵活造成带速不稳同样会使文件记不好，读不出。但是，用这些磁带录放音乐，语言节目时，也许你不会听得出这些毛病。

2. 如何用试听来鉴别录音机和磁带

首先，找来一盘高质量的原声音乐或语言带，然后，通过线路录音，转录到你准备用来录文件的磁带上试听新录的磁带，要求声音清晰、悦耳，不能有失眞和背景噪声，磁带运行平稳，不能时有忽快忽慢，产生类似跑调的效果，那么你的录音机就基本上可用了。

3. 录文件时的注意事项

记录文件时和录音机上音量、音调的位置大小沒有关系。但你必须把读插头由 H-01 机或录音机上拔下来，绝大部分录音机都会在读、写间构成某种迴路，严重影响记带质量。但反过来，当从录音机上读文件时，写插头不拔下来一般沒什么不良影响。

记带命令为 CSAVE“文件名”

文件名只允许1个字符（或汉字）。2个以上字符的文件名可能会引起读带错误，当你打完记带命令后（假设录音机已正常，先按下录音机录音键，让录音机运行几秒钟，在整个机器各部分都稳定好之后，再按 RETURN 键开始录音。当屏幕提示准备好之后，才能按停止键。你必须认识到，磁带机虽然便宜，但可靠性比磁盘差得多，每个文件至少存两

次，以避免不必要的损失。

4. 如何调整读带时的音量和音调

对于质量好的录音机，这是比较简单的工作，一般只要将音量开到足够大即可。音调低一些，音量相对要大一些；音调高一些，音量可以关小一些；即音调与音量要配合使用。对于输出功率小的录音机，可能要把音量开到最大，音调调到最高才能正常工作。但若输出功率足够，应该调低音调，增加音量，这样比较稳定。

对于某些性能较差或与H-01机配合不太好的录音机，可按如下步骤调整音调和音量。

1）输入AUTO↙（↙表示按RETURN）进入自动行号。按住'A'（或其它字符）不放，连续输入该字符，每满64个字符，自动进入下一行，共输入几十行。最后一行改为（设最后行行号为500）。

   500　　CSAVE "A"：GOTO 500↙

   然后按BREAK键退出自动编行状态，返回命令状态。

2）将录音机和H-01机按'写'磁带连结（注意：应拔下读带插头），按下录音键，并在H-01机上输入RUN 500↙，这样H-01机将反复地向磁带上写内存中的程序。10～20分钟后，按中断键，但机器不会马上响应中断，你应该按住中断键耐心地等到一次写入结束，然后停止录音。

3）把磁带转回到初始位置，插上读带插头，键入CLOAD "B"↙，按下放音键。

4）这时你可以反复调整音调和音量开关，直至H-01机发出的读带声十分平稳，没有断续，和写带时的声音提示相同即可，一次调整不成还可将磁带倒回再调。

5）调整好后，按复位键退回初始画面。不要再调整录音机，一般来说，录音机可以正常工作了。

如果按上述过程仍然不能把你的录音机调整好，这并不说明你的 H-01 机不能正常工作，你最好换一盘带，换一台录音机试试再说。

为使读写磁带文件顺利，你应该经常清洗磁头，定期作消磁处理，使你的录音机总处于最佳工作状态，你应该用优质磁带，并经常检查走带机构是否灵活等等。

4. 在磁带上存贮多个程序

通常在一盒磁带上足以保存几个 BASIC 程序，你应该注意把各个程序存放在不同的磁带位置上，绝对不能有重合。这和在磁带上存歌曲完全一样，你最好使用带有计数器的录音机，记下各个程序起始位置的计数值，便于以后查找。

第三章　运行程序的基本过程

H-01 机有4种操作状态：命令、执行、编辑和管理。在命令状态：计算机立即响应键入的命令。这是用来书写程序和直接计算的状态。无论何时，屏幕上出现>就表示处于该状态。在命令状态下，输入 RUN↲ 可进入执行状态，它使已经输入至机器内的 BASIC 程序得以执行。执行结束后，返回命令状态。

编辑状态允许你对内存中 BASIC 程序进行修改、添加和删除等操作。在命令状态下输入 EDIT 行号即可进入编辑状态。

在管理状态下，可将机器语言的目标文件输入到内存，或直接按给定的执行地址执行内存中的目标文件。

当屏幕上显示出>时，表示计算机处于命令状态，可以接受你的命令工作了。如果你键入3＋5，屏幕上只显示你键入的字符，即使你按了 RETURN 键也不能得到正确结果。这是因为计算机不懂你的意思，是显示键入内容呢？还是要计算结果。如果要计算结果，在算式前应键入一个命令 PRINT 或？（相当于 PRINT），变成？3＋5，再按 RETURN 键，显示出计算结果8，这就是将计算机当作计算器使用。你还可以变化一下输入？“3＋5＝”；3＋5↲，将显示出3＋5＝8。这是引号起了作用。你不防写一些更复杂的算式试试，要注意：乘号用＊、除号用／表示。机器遵守先乘除后加减的运算次序。

但是，如果你要解决一个复杂问题，需要许多句话（在计算机中叫语句），你可以把许多语句组织起来构成一段程序。一个程序行内可写几个语句，中间用冒号分开，每行用标号开头。标号必须是正整数，每个标号必须是唯一的，两行程序不能有相同的标号。如果多次应用了相同标号，则计算机只记住最后送入的那行程序，标号小的程序行在前面，标号大的排在后面，即使各行不按顺序打入，计算机自动会按标号大小顺序排列的。

在 H-01 机的一个程序行中，对语句的数目没有特殊限制，但一行的总字符数不能多于64个，否则将自动进入下一行，即在第64个字符后面出现提示符>。

如果你在每行前面用了标号，输入完一行，按回车键，计算机并不马上执行，而是将光标移到下一行起始位置，直到全部程序都送进机器，并接受执行命令 RUN 时，它才进入执行状态，并顺序地边检查边执行。

例：

```
10  A=3:B=5:C=A+B↙
20  ? A,B,C↙
30  END↙
RUN↙
```

屏幕显示　3　　5　　8

注意：↙表示按 RETURN 键。

假如你现在想算 A*B，你应进入编辑状态对内存中程序进行修改，此时，你应该修改第10行，输入 EDIT 10↙便进入编辑状态。H-01 机每次只能对一个程序行进行修改。如果你要同时编辑第10行和第20行，必须在修改完第10行后退回到命令状态，再输入 EDIT20↙去修改第20行。

下面给出4种状态间的转换图（图3—1），我们将在后

面章节里逐一介绍。

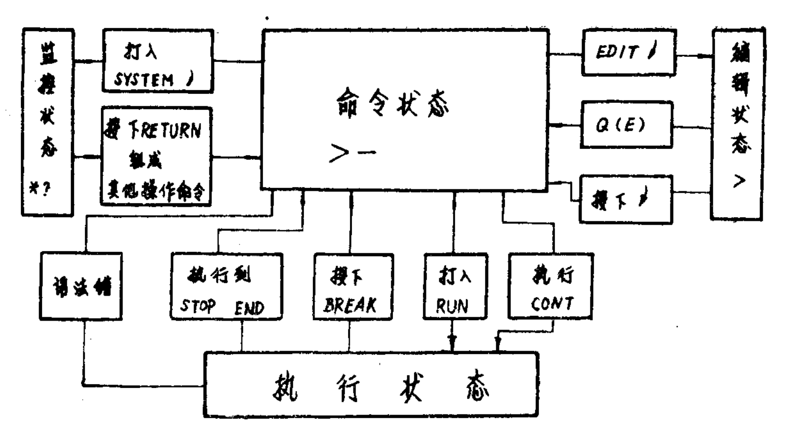


图3—1 BASIC Ⅱ 四种操作状态转换图

第四章　命令、编辑状态

§4.1命令状态

1．命令状态下的特殊功能键

1）RETURN键

用作回车，计算机'注视'刚输入的一行内容。如果沒有语句标号，计算机将对全行所有语句进行解释执行。

2）BS键 "←"

光标退格，并删去刚打入的字符。

3）@键

暂停程序的执行，按任何键（个别特殊功能键除外）能使执行继续进行。在LIST过程中，文本是不断向上移动着的，按该键可停止上移（即所谓'冻结'），便于检查。

4）ESC键

停止执行，即中断。要继续执行时须打入CONT。

2．命令

在本节介绍控制计算机的命令，其中绝大部分(除CONT外）也可以写在程序中作为语句，这在某些情况中是很有用的。

1）| AUTO　起始标号，增量 |

该命令打开一个自动编标号的功能，使用户不必在输入程序时为每一行语句输入标号，只要输入实际的程序语句。命令中增量表示相邻两行语句标号之差。若只输入 AUTO↲那末语句标号由10开始，增量为10，每按一次回车，计算机自动结束本行，显示下一行的语句标号。要停止执行 AUTO 功能，按 ESC键。

当 AUTO 给出一个已经被使用的标号时，会在该标号旁出现一个‘*’号，这时，必须退出 AUTO 功能去处理，或者删除已有的行，或者重新指定一次 AUTO 的起始标号和增量。

2) CLEAR n

当不带自变量 n 时，这个命令把所有数字变量、字符串变量清零。带自变量 n 时，它同时还分配 n 个字节作为字符串存贮单元。在开机时，计算机自动执行 CLEAR 50。

3) CLOAD "文件名"

命令从磁带机上输入一个 BASIC 文件到计算机内存中去。在本命令及下面的 CSAVE，CLOAD? 命令中，文件名只允许为一个字符或汉字。若忽略文件名，计算机只把遇到的第一个程序输入。若使用了文件名，计算机只输入该文件名的程序，而对其它程序不予理睬。

从磁带上输入一个程序，将自动清除以前存贮在计算机中的程序。

如果使用了磁带上沒有的文件名，或由于录音机、磁带质量等问题造成读带失败，必须用 RESET 键复位，使计算机退出死锁状态。

为避免稍有错误便使全部程序无法输入，H-01机只要能正确地读入程序的起初部分和结束部分，就把中间读到的内容输入到内存，然后返回命令状态。如果你已经调好的程序在读入后运行出错，你必须修改，或者重新调整一下录音机，再读一次（录音机调整参见第二章第九节）。

4） CLOAD？“文件名”

此命令比较存贮在磁带上的程序和内存中的程序。当你往磁带上转贮（见CSAVE），并希望核对转贮是否成功时，使用这个命令。如果在CSAVE中使用了文件名，则应和用命令CLOAD？“文件名”冒号中的文件名必须相同。若使用CLOAD？命令而未加“文件名”，则只核对遇到的第一个程序。带上的程序和内存中程序一个字节一个字节的比较，若有差异（指转贮不好），将给出信息“坏”，这时你应该重新转贮。与CLOAD不同，CLOAD？不清除内存。

1） CONT

当程序执行过程中产生暂停时（按了ESC键或遇到程序中的STOP语句），输入该命令可由停止处继续往下执行。在暂停期间允许检查变量的值（用PRINT）或修改变量值，然后输入本命令，程序将以新的变量值继续进行。主要用于调试程序。

注意：用EDIT或其它方式改变程序后，不能使用CONT。正常执行结束，CONT也无效。

6） CSAVE “文件名”

将机内程序转贮到磁带上去。这一命令必须有规定的文件名。文件名可以是除双引号外的任意字符或汉字。但只能

使用一个字符或汉字作文件名，在转贮完毕后，最好能将磁带倒回起始位置，再用CLOAD? 核实一次。

由于磁带可靠性比磁盘差，为了使你宝贵的程序或数据不被丢失，建议你至少将每次转贮的内容存两次。

7) **DELETE 语句标号—语句标号**

从内存中删除一行或一段程序。

DELETE	语句标号	删除标号标识的程序行。
DELETE	标号—标号	删除两语句标号之间的所有程序行。
DELETE	—标号	删除由头开始至指定标号前的所有程序行。
DELETE.		删除现行行（指刚输入或编辑过的行）。

8) **EDIT 标号**

进入编辑状态。（详见§4.2）。

9) **LIST 标号—标号**

命令计算机显示内存中的程序行。

如果只输入LIST，则全部程序将连续显示出来。

LIST	标号	显示指定程序行。
LIST	标号—标号	显示两标号之间的所有程序行。
LIST	标号—	显示由标号至末尾的所有行。
LIST·		显示现行行（即刚输入或编辑过的行）。
LIST	—标号	显示由头至指定标号的所有

程序行。

LLIST：功能与 LIST 相同，只是把程序清单输出到打印机中去。

10) NEW

本命令清除内存中所有程序。清除屏幕显示，置数字变量、串变量为零。但它不改变由前面 CLEAR n 所规定的串空间。

11) RUN 标号

命令计算机执行内存中的程序。若不规定标号，从头（最小行号）开始执行。若指定语句标号，由该语句开始执行。

RUN 可以用作程序中的一个语句，对于象游戏等循环程序，这是一个重复地清除和开始运行的方便办法。

12) SYSTEM

命令计算机进入管理状态。它允许你输入由本机编辑/汇编程序所产生的目标文件。

输入完 SYSTEM 后，显示*?，你可直接输入文件名（不用引号，当输入完后，显示另一个*?，这时输入一个'/'，随后输入启动运行地址（十进制），输入回车后，计算机便运行目标文件。

使用 POKE 语句打入内存的机器码也可由此办法运行，但要在第一个*? 后输入/启动地址。

13) TROFF

关闭执行跟踪功能。参看 TRON。

14) TRON

打开执行跟踪功能，使你跟踪程序流程来分析程序的执行情况。当计算机每执行一条程序行，其语句标号就显示在一对括号里。

TRON 和 TROFF 可以写在程序中，以便帮你了解某些特定语句的执行情况。

15) MODE (0)

命令进入中西文兼容的显示方式。

16) MODE (1)

命令进入全西文方式。

17) CLS

清屏幕，但不清除内存。

§4.2 编辑状态

本节介绍 H-BASIC 的编辑功能，它允许你对内存中的一行 BASIC 程序进行修改，添加和删除。

下面分别描述编辑命令、子命令和特殊功能键：

1. EDIT 语句标号

这个命令使你进入编辑方式。（注：必须指定要编辑的语句）。它有两种方式进行编辑：

EDIT 语句标号↲　　让你编辑指定行。若没给出语句标号，将产生FC错误。

EDIT.↲　　让你编辑当前行。即刚输入

或修改的一行，这行是发生错误的行。

例如语句：

100　FOR　I=1TO10　STEP　2:PRINT　I:NEXT

打入 EDIT　100　↲，则有：

100—

于是，你即可开始编辑行100了。

2. RETURN键

在编辑方式时，按 RETURN 键（↲ 键），将使计算机执行你对本行所作的修改并回到命令方式。

3. n 空格键

将光标向右移动 n 格。在编辑方式中，敲空格键可使光标向右移动一格，并显示出前面位置上存储的字符。若一次需移过 n 格，首先打入格数 n，然后敲空格键。

4. n←(退格)

把光标向左移 n 格。并在左移过程中把前面的字符都从显示器上擦去，但并没有从程序中删除。若不指定数目 n，光标只退一格。

5. SHIFT ^

同时按下 SHIFT 键和 ^ 键，可从任何插入子命令（X，I 和H）中退出，在退出一个插入子命令后，仍处在编辑方式，光标仍在原位置上。

6. L (开一行清单)

在计算机处于编辑方式且还未执行一个插入子命令时，按下L键可使程序行的剩余部分显示出来，并且在显示器的下一行重新给出该行的语句标号，光标移到该行的首位。

主要用在编辑期间，观看本行目前的内容。

7. X （移到行的末尾和插入）

使本行的剩余部分显示出来，光标移到行末尾，并使计算机处在插入子命令状态。于是你能在行末尾增加内容。

8. I （插入）

允许在本行中从光标前位置起插入某些内容。

9. A （取消修改并重新开始编辑

把光标移回到程序行的开头，并取消所做的修改。例，你若在行中添加、删除或修改了一些内容，且又希望回到这一行的开头并取消已做的修改，则你先按下SHIFT ∧ 键，然后按下A键，光标将下移一行（语句标号不变），并移到该行首。

10. E（保存修改和退出编辑方式

使计算机结束编辑并保存已做的全部修改。

注：按E键时必须在编辑状态下。

11. Q （取消和退出）

使计算机结束编辑并取消这一编辑期间已做的全部修改。若你决定不修改程序了，按下Q键就取消了修改并退出编辑方式。

12. H （切尾和插入）

使计算机删除该行程序中光标后的剩余部分，并可以从光标的目前位置起插入某些内容。

13. **nD （删除）**

使计算机从光标的右边删除指定数目 n 个字符。被删除的内容用两个警叹号（!）相夹。

14. **nC （修改）**

使计算机由光标位置起修改指定数目 n 个字符。不指定 n 时，只修改 1 个字符。

15. **nSc （检索）**

使计算机检索字符 c 第 n 次出现的地方，并把光标移到这个位置，若不指定 n 时，则检索第 1 次出现的指定字符c，若 c 未找到，光标将移到程序行的末尾。

16. **nKc（检索和"抹除"）**

使计算机删除字符 c 第 n 次出现的前面的所有字符，并把光标移到这个位置。

第五章　H-BASIC语言

§5.1　基本成分

1.　变量名称

变量名称必须由一个字母（A-Z）开头，后边可以跟随另一个字母或一个数字（0～9），因此以下都是正确的变量名称。

A、A2，AA，AZ，ZZ

变量名称可以多于两个字符，但计算机仅识别前两个，如SU和SUM被看成同一变量。

注意：不能使用BASIC语言中具有特殊意义的词汇（也不能包含）作变量名称。不能用作变量名称的全部"保留词汇"在附录1中给出。

2.　变量类型

H-BASIC中有四种变量类型：整型、单精度型、双精度型和字符串型（简称串型），只要简单的在变量名后面的一个标识符便可。下边是全部清单。

3　数组：

数学中有矩阵，它可以简单地表示了一组有一定排列顺序的数值。H-BASIC也是一样，应用数组表示一组有一定顺序的数字或字符串。数组当然是由同类型的数组元素组成，根据数组元素之类型，可分为：整型数组，单精度数

变　量　类　型	标识符	例　子	存入的型数值
整型（大于－32769小于＋32768的全部数字）	%	A%，B%	－30，123，5001
单精度型（6位有效数字）	!	A!，AA!	1，3.14159，－50.12
双精度型（16位有效数字）	#	A#，ZZ#	－300.12345678
双精度科学表示法（在输入常数或输出很大或很小的数时）	D	A#＝1.2345678901 D＋12	1.2345678901 $\times10^{12}$
串型（最多128个字符）	$	A$，HP$	"ABCD" "中国科学" "1＋2＝?"

不带标识符就认为是单精度型的，A%。A!，A#分别代表不同变量的。

组，双精度数组和字符串数组。

数组中的元素都带有下标，称为下标变量。例如：S%(I,J) 表示I行J列的整型S数组的元素。下标变量中下标的个数称为数组的维数。

程序中要用数组时，一般需要对该数组先进行定义，以便让计算机给该数组分配相当的存储单元。定义数组用说明语句：

格式：DIM<数组表>

将为每个数组定义类型，维数及每个维的下标上界值。

例如10 DIM A(4)，B(5,4,2)，C$(20)

其含义为定义一维数组A，下标最大值是4，即从0-4；三维数组B下标分别从0-5，0-4和0-2；一维字符串数组C，下标从0-20，共21个元素，每个元素不超过64字符。如果沒有

予先定义A和B的类型，那么A和B被认为是单精度数组。

DIM语句可放在程序中的任何位置，但必须放在引用该数组之前。

一个数组能够具有多少维数（大小及长度）要受可利用内存大小的限制。

例如：统计30个学生的成绩，得100分的有多少，得90-99分的有多少，……30—39分有多少。可编程如下：

```
10  DIM A(7)              (分8类，初值为0)
20  READ X
30  IF X=0GOTO 70         (当X =0时去打印结果)
40  H=X/10—3              (确定下标变量H)
50  A(H)=A(H)+1           (使下标变量内容加1)
60  GOTO 20
70  FOR I=OTO 6
80  PRINT 10* (I+3) ; "--" ; 10* (I+4)-
    1, A(I)
90  NEXT I
100 PRINT 100, A(7)
110 DATA 85,76,61,68,93,100,94,73,79,80,89,
         67,58,79,81,98,67,99,75,81,91,80,
         70,65,67,72,81,86,68,71,0
RUN
  30—39    0
  40—49    0
  50—59    1
  60—69    7
  70—79    8
```

80—89　　8

90—99　　5

100　　1

4. 算术运算符

H-BASIC 的算术运算符有：

+(加)，-(减)，*(乘)，/(除)，[(指数，例2[3 = 8)

运算次序：→ ( ) → 函数 → 幂 →

*、/ → +、- →

5. 关系运算符

< (小于)，> (大于)，= (等于)，<> (不等于)，<= (小于或等于)，>= (大于或等于)。

6. 逻辑运算符

AND (与)，OR (或)，NOT (非)。

7. 字符串算符

< (小于)，> (大于)，= (等于)，<> (不等于)，<= (小于或等于)，>= (大于或等于)，+ (连接)。

8. 运算的优先级数

首先执行最内层括号中的运算，然后一层层往外执行。同一括号中的运算按下面优先级别执行：

[(指数：A[B)

- （取负值：-X）

*，/（同一优先级数，运算自左至右，下同）

+，-

<，>，=，<=，>=，<>

NOT

AND

OR

9．缩写：

?　作 PRINT

'　作 REM

.　代表当前行。在打印清单，编辑或删除命令中代替当前行的语句标号。

10．错误信息：

H-BASIC 使用一组错误代码（见附录5）给出关于错误类型的信息，也指示了犯错误的程序的语句标号，以便寻找程序中的错误。

也有一些难于发现的错误：比如你在输入程序中按了光标上移键，又按了光标下移键，这一行程序在屏幕上看起来是不错的，但根本不能执行。所以，如果对一行给出语法错误信息的语句，找不出任何错误的话，就应该重新输入这一行。

§5.2．输入/输出语句

本节所述语句使你能在人⟷计算机⟷打印机⟷显

示器←→录音机之间交换信息，它们在程序中用于输入数据，输出结果。

1. PRINT 输出项目表

在显示器上输出项目表中的项目。项目表可以由一个或多个项目组成。项目可以是数字常数、变量、串常数（用引导引起来）串变量，也可以是由它们组成的表达式，各项间用逗号或分号隔开。使用逗号，分区打印，使用分号，在两项目之间不插入空格，故又称作紧凑格式。

例如：

```
50  x=5
100 ? 25; "IS  EQUAL  TO" ; x[2
```

运行结果：

```
␣25␣IS  EQUAL  TO  25
```

又：

```
10 A$ = "字符串"
20 ? A$; A$, A$; "␣"; A$
```

运行结果：

```
字符串   字符串␣␣字符串␣字符串
↑                 ↑
0                 14
```

打印出的正数前面有一空格，（代替+号），所有数字后面有一个空格。字符串的前后部分都不插入空格，引号中的內容，原样不动输出，允许你在引号中使用任何符号，比如汉字、英文字母、自造字或第二字符集中的字符，而在其它场合，这些都是违法的。

又如：? "ZONE1"， "ZONE2"， "ZONE3".

运行结果

ZONE1　　ZONE2　　ZNOE3

↑　　　　↑　　　　↑

0　　　　14　　　　28

可见每行分 3 个区，每区14个字符，每个汉字占两个字节位置。

又：10　？ “区1”，“区2”

结果

区1　　　　　区2

↑　　　　　　↑

0　　　　　　14

每迁到一个逗号，就将移到下一个打印区。

用分号结尾的使本行项目表全部输出后不换行，可以和下行显示结果连在一起，不用分号结尾，本行执行结束后自动转入下一行。

LPRINT 输出项目：与 PRINT 功能相同，只是将输出内容输出给打印机。

2. PRINT@位置，项目表；

可以精确地指定开始显示的位置，在 MODE(0)时，规定的位置应该是0～461；在 MODE(1)时，规定的位置应该是0～671。位置可以用常数，变量或表达式给出；项目表同 PRINT 中的说明。

3. PRINT TAB（位置）项目

PRINT@中的位置从第一行第一个字符算起，而本语句中的位置由当前行的首字符算起。TAB 可以在一条 PRINT

语句中多次使用。

例如 10 PRINT TAB(5) "TABBED␣5"；TAB(25) "TABBED␣25".

运行结果：

```
␣␣␣␣␣TABBED␣5          TABBED␣25
     ↑                  ↑
     5                  25
```

表达式的值必须在0～255之间。由于每行只有42个字符位置，如果你指定的位置大于41，光标将移到下行，后面的TAB也相对于新行来计算位置。

例如：10 PRINT TAB(5) "TABBED␣5"；TAB(50) "TABBED␣50"；TAB(20) "TABBED␣20"

运行结果：

```
␣␣␣␣␣TABBED␣5
     ↑
     5
␣␣␣␣␣␣␣␣TABBED50    TABBED␣20
        ↑           ↑
        8           20（新行）。
```

可以用数学表达式来指定TAB的位置，这在利用TAB来做数学函数的曲线、图表时非常方便。

TAB不能用来使光标往左移动，所以后一个TAB的位置数一定要比前一个大。如果光标超过指定位置，TAB将不起作用，仅相当于分号（；）。

4. PRINT USING 字符串：项目表

该语句用于指定数字变量和字符串的打印格式。主要用

于打印标题、帐单、发票或者需要特定打印格式的场合，打印格式的说明字段放在字符串中，可以是字符串常数或变量。

在字符串中可使用的格式说明字段如下：

#	规定打印位数，一般说明形式为##…#·#…#，如果说明字段的整数部分比要显示的数的位数多，左边用空格补齐；比实际要输出的位数少，将输出格式溢出提示符%。如说明字段的小数部分比要输出的小数部分位数多，右边以0补齐；反之，将小数部分4舍5入后按说明的位数输出。如果没有0数点及小数部分，将只输出整数部分。 在第一个#号后和小数点之间的任意位置放一个逗号，将使输出数字的每三位之间出现一个逗号，逗号在字段中占1位。
**	放在说明字段的开头，将使左边多出的位数用*号而不是用空格补齐。
$$	放在说明字段的开头，将使数值前面出现一个美元号，左边多余位用*号补齐。
+	加号可放在说明字段的前边或后边，按其所在位置，正数输出+，负数输出-。
-	放在说明字段后面，所有负数后边跟一个负号，正数后为一个空格。
%␣␣%	用于打印多于一个字符的字符串。允许打印的字符串的字符数为符分号之间的空格数加二。
!	表示打印字符串的第一个字符。
[[[[	表示指数格式，附在###·#后边。

说明字段　　输出项目　　输出结果

##.##	12.12	12.12
###.##	12.12	␣12.12
##.##	121.12	%121.12
##.###	12.12	12.120
##.#	12.12	12.1
##	12.12	12
**###.#	12.12	***12.12
$$##.##	12.12	␣$12.12
**$####	12.12	****$12
+###.#	12.12,-12.12	␣+12.12, ␣-12.12
###.#+	12.12,-12.12	␣12.12+ ␣12.12-
##.##-	12.12,-12.12	12.12␣, 12.12-
%␣␣%	"ABCDEF"	ABCD
%␣␣%	"中国科学院"	中国科学
!	"ABCD"	A
#.###[[[	123.45678	0.123E+03
##.###[[[	123.45678	1.234E+02
###.####[[[	0.0000005678	56.7800E-08

如果在说明字段后面是若干项目制成的表，则所有项表都按说明字段指示的格式输出。

5. INPUT 项目表

使计算机暂停执行，等待你通过键盘给项目表中的项赋

值。表中各项必须用逗号隔开，输入数值的类型必须与语句中规定的类型一致。例如，不能将一个字符串值输入给一个数字变量，否则计算机会提示“沒有数据”。

例如　100 INPUT X$，X1，Z$，Z1

这语句要求你的输入次序为：字符串，数值，字符串，数值。当执行到该语句时，显示问号‘？”。你可按下列方法赋值：

“张三”，50，“JACK”，40↵

（输入字符串时两边的引号是必不可少的）。也可输入一个值，按一次回车，计算机提示两个问号来等待下面输入。

如果输入项多于语句中所规定的个数，计算机会告诉你“多余项被略去”。

在INPUT语句中也可包含提示信息，减少输入错误，比如：

100　INPUT“人名␣年龄␣性 别”；A$，B，C$

注意：只能把数字常数输入给数字变量；字符串常数输入给串变量。

6. READ　变量表

该语句必须和DATA语句配合使用，参见DATA语句，

7. DATA　数据表

将置数据语句（DATA）和读数据语句（READ）联合使用可以给变量赋值。计算机将DATA语句的数据依次送入数据区。当执行READ语句时，就从数据区中依次赋给每一个变量。

由于DATA并不产生任何操作，它是一个非执行语句，因而只能放常量，而不能放表达式，而且它可以放在程序的任何位置，而READ语句是一个执行语句，它一般放在需要使用该变量的位置。变量个数，只能少于或等于数据个数。

8. RESTORE

RESTORE语句是恢复数据语句，执行该语句后，允许多个READ语句使用相同的DATA语句。读取位置定位在第一条DATA语句中的第一个数据项。

9. PRINT #-1，项目表

把项目表中的内容记录在磁带上。（当执行这一语句时，必须将磁带机连接好，并处于录音状态。）

例如 PRINT #-1, A1, B$, "THAT'S ALL"

这样就把A1，B$中的值及串常数"THAT'S ALL"记录在磁带上。

特别注意：项目表中包含的总字节数不能超过200个，否则会产生不可予期的错误。因此，如果项目表太长，应该分成两条或多条PRINT #-1语句。

10. INPUT #-1，项目表

从磁带上读入规定的数值并赋给指定的变量（执行这一语句时，必须连接好磁带机并处于放音状态）。INPUT#-1，的项目表和PRINT #-1的项目表在项目个数和类型上完全一致。

如果INPUT #-1的项目表中的项目少于PRINT #-1记在带上的项目，计算机将提示"多余项被略去"；如果情

况相反，计算机将不断查找。只有在正确地读完INPUT ＃-1所需全部项目或按复位键才能退出查找，为保护在内存中已有的程序，你应特别当心INPUT ＃-1的使用，在类型不符时计算机会提出 FC 错误。

§5.3 程序语句

本节对 H-BASIC 的程序语句逐一进行解释。

1. DEFINT 字母范围

定义某个范围内任何以字母开头的变量都是整型变量。除非在变量名称后附加别的类型说明符，再按所说明类型存贮和处理。整型变量的定义范围是－32768～＋32767。

例如：DEFINT A，I，N

2. DEFSNG 字母范围

定义某个范围内任何以字母开头的变量为单精度变量。除非变量名后加有其他类型的说明。单精度的变量和常数以 7 位精度存储而以 6 位精度输出。

例如 DEFSNG I，W-Z

3. DEFBL 字母范围

定义某个范围内的任何字母开头的变量为双精度变量。除非加有其它类型的说明符，双精度允许以17位精度存储，但被输出时只显示16位。

例：100 DEFDBL S-Z，A-E

4. DEFSTR 字母范围

定义某范围内的字母开头的变量为串变量。除非变量名后加有其他类型的说明符。若已经因 CLEAR 开辟了足够的串存储空间，那末每一个字符串最多能够存储128个字符。

例如 10 DETSTR L-Z

5. CLEAR n

为计算机开辟 n 字节的内存用以存储字符串。另外使所有变量清零，一开机，自动地为字符串开辟了50个字节的内存。其中 n 可以是常数或表达式。

用 CLEAR 开辟的字符串存储空间必须等于或大于字符串变量在执行过程中需要存储字符的最大数目，否则将产生 (OS) 错误。

例 100 CLEAR 1000

这样就为字符串存储开辟了1000字节的内存空间。

6. DIM 数组名称（维1,维2,……,维 K)

为一个数组或一组数组设置“长度”（每一维允许的元素数目）。如果不用 DIM 语句，那么每个数组的每一维允许使用的长度为11（下标0～10）。

例如：DIM A(5), B(2,3), C$(20)

7. LET 变量=表达式

用来对某一个变量进行赋值，即计算右边的表达式，并将结果赋给左边的变量。在 H-BASIC中，LET 可写也可不写。

例如：10 LET A$ = “H—01机”

20　B = 1.23

注：赋值号"="与数学中的等号"="具有不同的含义。A = A + 1数学上是不成立的。

8. END

使计算机结束程序的执行，并返回命令状态。END主要用来强制计算机在某处结束执行，而不是程序的逻辑结尾。

9. STOP

该语句使计算机暂停程序的执行，并在屏幕上显示"中断在<语句标号>"。调试程序时作为手段，用以检查或修改变量的当前值，打CONT可使程序从停止地方继续运行下去

10. GOTO 语句标号

这是无条件转向语句。它使计算机从正常的执行顺序中，转向需要执行的程序行。一般，计算机按行号从小到大，行内从左到右的顺序执行。

例如：100　GOTO　10

当执行100行时就控制无条件转向10行。

例
```
10  CLS
20  T = 0
30  INPUT "单项值 = "; X
40  T = T + X
50  ? "累加为"; T
60  GOTO 30
```

11. GOSUB（转子语句）和RETURN（返回语句）

在某些计算问题中，往往需多次运用一段程序，为避免

程序重复，可把这段程序编成相对独立的“子程序”。在主程序执行过程中若需要这段程序时，可使用转子语句（GOSUB）调用子程序；子程序执行完毕由返回语句（RETURN）返回主程序。

例：计算5！+10！+15！。

编程如下（将n！编成一个子程序）

```
10   N=5                  （将5赋给N）
20   GOSUB  110           （转子，计算5！）
30   X=K                  （把5！值暂放X中）
40   N=10                 （把10赋给N）
50   GOSUB  110           （转子，计算10！）
60   Y=K                  （把10！值暂放Y中）
70   N=15                 （把15赋给N）
80   GOSUB  110           （转子，计算15！）
90   PRINT  X+Y+K         （打印5！+10！+15！值）
100  END
110  K=1
120  FOR  I=1  TON
130  K=K*I
140  NEXT
150  RETURN
160  END
```

12. ON 表达式 GOTO <一组标号>（选择转向语句）

这是一个多分支的转移语句，它受控于一个表达式。表达式的值必须是0～255内。当计算机执行该语句时，先对表达式的值取整，然后在GOTO 后面找到相应的项，并转移到

该项所指的语句标号去。

例　100 ON X GOTO 150,160,170,180

其意义是：当X＝1，转向语句150

X＝2，转向语句160

X＝3，转向语句170

X＝4，转向语句180

例如计算：

$$Y=\begin{cases}2x+3 & x<0\\ 0 & x=0\\ 2x+5 & x>0\end{cases}$$

编程：

```
10  INPUT "X=" ; X
20  ON SGN(X)+2 GOTO 30,40,50
30  PRINT "Y=" ; 2x+3: GOTO 10
40  PRINT "Y=" ; 0   : GOTO 10
50  PRINT "Y=" ; 2x-5: GOTO 10
```

注：符号函数为－1,0,＋1，加工后值为1 2 3。

13. **ON <表达式> GOSUB <一组子程序行号>**

该语句是选择转子语句。功能类似于选择转向语句（ON-GOTO），只是控制转移到一个由语句标号清单规定的某个子程序去。表达式规定同上。

14. **FOR 变量名称＝表达式 TO 表达式STEP 表达式 NEXT 变量名称**

这是一个循环语句。程序进入一个循环圈，使得一段程

序语句重复执行规定的次数。其一般格式是：

语句标号　FOR 计数变量＝初值 TO 终值[STEP 步长]
　　　　　⋮　[程序语句]　　（循环体）
语句标号　NEXT [计数变量]

在FOR语句中，初值、终值和步长可以是常数、变量或表达式。循环语句的执行步骤是：

1）求在FOR语句中三个算术表达式的值（初值，终值，步长），存储起来。

2）将初值赋给计数变量。

3）执行循环体语句。

4）遇到NEXT 语句时，修改计数变量值，并检查是否超过终值。（当步长为正值时，指大于终值；步长为负值时，指小于终值）。若超过终值，则转去执行 NEXT下边的语句，否则，继续执行循环体语句。

例：求100～999之间的全部水仙花数。

水仙花数是三位数字等于它每一位数的立方和。如：$153=1^3+5^3+3^3$

编程如下：

```
5   CLS: ? TAB(5) "求水仙花数"
10  FOR I=100 TO 999
20  X=INT(I/100)
30  Y=INT((I-X*100)/10)
40  Z=I-X*100-Y*10
50  IF I< >X*X*X+Y*Y*Y+Z*Z*Z
    THEN 70
```

```
60  PRINT I,
70  NEXT I
80  END
RUN
       153     370     371     407
```

15. ERROR 代码

这是一个“模拟”错误语句，其格式：ERROR<错误代码>当碰到ERROR代码语句时，计算机将精确地按这种错误已发生时那样处理。参见附录 5：错误代码及其意义。

例如

```
100 ERROR 1
RUN
? NF ERROR
READY (准备好)
>-
```

错误代码 1 是“没有匹配的FOR语句，试图执行 NEXT 语句。”

16. ON ERROR GOTO 语句标号

当计算机在程序中碰到任何一种错误时，通常它中断执行并印出错误信息。而当使用 ON ERROR GOTO 语句标号语句时，使你能够设置一个捕获错误程序，它允许你的程序因错误而“再生”，并继续执行而不中断。一般，使用 ON ERROR GOTO 语句时，你已经考虑到会产生某一种特殊类型的错误。

例如：

```
5   ON ERROR GOTO 100
10  C=1/0
```

在这个明显错误的程序中，当计算机试图执行语句10时，就会产生被零除的错误，但由于语句5，计算机就转移到100开头的错误处理程序中去执行。

17. RESUME 语句标号

结束错误处理程序，使程序返回到指定的语句恢复正常执行。若无语句标号或是零，计算机将回到产生错误的那条语句。RESUME NEXT 将使计算机转移到产生错误的那条语句下边一条语句上去执行。

例如：

```
5   ON ERROR GOTO 100
10  INPUT X
20  PRINT X[2
30  GOTO 10
100 PRINT SQR(X)
110 RESUME 10
```

18. REM（注释语句）

REM 是一个解释语句（非执行语句），用于在程序中插入注释以对程序加以说明，便于阅读源程序。

19. IF……THEN……ELSE（条件语句）

这是一个条件语句。它让计算机根据给定的条件，经过分析，比较和判断，然后按不同情况作出不同处理。汉化即是“……就……否则……”。

格式有： IF E THEN E_1

IF E THEN E_1 ELSE E_2

即若E为真，则执行 E_1，否则是一个行号也可是一个或

多个语句。

例如：

```
5    CLS: ? TAB(3) "请小朋友做加法"
10   A=RND(99): B=RND(99): N=0
20   ? A; "+"; B; "="
30   INPUT C
40   IF C=A+B ? "对了，请做下一题":
     GOTO 10
50   ? "错了。重做"
60   N=N+1: IF N<3 THEN 20 ELSE 10
      (计数次数)
```

§5.4 字 符 串

1. 字符串输入/输出

字符串输入完全可以象数字常数那样，使用INPUT READ/DATA和 INPUT#-1(由盒式录音机输入)，输入给程序，同理，亦可以用PRINT，PRINT#-1等语句输出，使用LET语句，可以为一个字符串变量赋值。字符串常数在任何时候都必须用引号括起来。

2. 字 符 串 比 较

字符串的比较是从左至右逐字节进行的，并规定：ASC值较小的字符比ASCⅡ值较大的字符"小"。这样，我们可以使=，< >，<，<=，>，>=等关系运算符来比较字符串。例如："A!"<"A#"。因为"! "的ASCⅡ值为33，"#"为35。长度不同的字符串比较时，短的比长的

"小"。例如："A—" > "A"

3. 字 符 串 运 算

字符串只有一种叫做"连接"的运算。用"+"号表示。例如：

```
10  A$ = "A ROSE"; B$ = "是一朵瑰玫"。
20  C$ = A$ + B$
30  PRINT C$
```

运行结果为：A ROSE是一朵瑰玫。

4. ASC（字符串）

求括号中字符串第一个字符的ASCⅡ值（以十进制表示）。若串变量为0，产生错误。例如：

```
10  PRINT ASC ("A")
20  T$ = "AB"; PRINT ASC (T$)
```

两语句得出同样的运行结果：65（A的ASCⅡ码）

5. CHR$（表达式）

执行ASC函数的反转换。给定一个ASCⅡ值，求与之相应的字符。自变量可以是0～255之间的任何数字，或表达式。自变量必须放在括号内。例如：

```
10  FOR I=32 TO 219
20  PRINT CHR$ (I);
30  NEXT
40  END.
```

6. FRE（字符串）

当把一个串常数或串变量作自变量时，FRE得到现在用于字符串存贮的空间大小。例如：

```
500 PRINT FRE (A$) ; FRE (L$) ;
FRE ( "Z" )
```

都得到相同的值，因为括号中使用的串变量没意义，是一个虚拟变量。（注：FRE求得的串存储的内存空间数量，完全取决于命令CLEAR n，不论FRE的自变量是什么，FRE（串）总等于n。）

7. INKEY$

在执行该函数时，计算机扫描键盘一次。如果在扫描瞬间没有按键，那么得到零串，程序继续执行下去。这是唯一的输入字符串常数不用加引号的地方，这是一个非常有用的功能。因为它允许不用回车键输入字符。在编制游戏程序。例如高炮射击、走迷宫等，都需要INKEY$ 功能。给INKEY$ 输入的字符并不在屏幕下显示出来。可以把INKEY$ 放在循环里，使程序反复搜索键盘，例如：

```
10  IF INKEY$ = " " THEN GOTO 10
20  … …
```

运行该程序时，如果你什么键也不敲，就在第10句循环，一直等到某键按下才转入下一句。语句IF INKEY $ = " " 是将一次读键盘结果与零串比较。

8. LEFT$（串，n）

求字符串的前n个字符。串可以是串常数或串表达式，n

可以是数学表达式。例如：

```
10  A$ = "TIMOTHY"
20  B$ = LEFT (A$, 3)
30  PRINT B$; "-- THAT,S SHORT FOR";
    A$.
```

运行后，屏幕显示为

TIM — THAT,S SHORT FOR TIMOTHY

9. LEN（串）

求括号中串的长度。（即由多少个字符组成）。串可以是串变量，串表达式或串常数。例如：

```
10  A$ = "  "; B$ = "TOM": C$ = "字符
    串"
20  PRINT A$, B$, C$
30  PRINT LEN (A$), LEN (B$),
    LEN (C$), LEN ("AB汉字")
        TOM       字符串      AB汉字
0       3         3          4
```

10. MID$（串，P，n）

求字符串的子串，其中长度为n，由第P个字符算起。串是常数或串表达式。P，n可以是常数或数字表达式。例如：

MID$（L$,3,1）得到串L$的第3个字符。如果在函数中不指定长度n，则得到由P开始的全部字符。

例如：

```
10  A$ MID$ ("ABCDEFG", 4)
```

```
20  PRINT A$
```
运行结果为：DEFG

11. RIGHT$（串，n）

求字符串的后n个字符。如果n>=字符串长度，得到全部串。

例如：
```
10  A$ = "ABCDEFG"
20  B$ = RIGHT$ (A$, 3)
30  PRINT B$
```
运行结果是：EFG

12. STR$（数学表达式）

将数学表达式或常数转换成字符串。例如：
```
10  A=58.5: B=-58.5
20  PRINT  STR$ (A), STR$ (B),
    STR$ (A+B), STR$ (A) + STR$ (B)
```
运行结果：

␣58.5 -58.5 0 ␣58.5-58.5

13. STRING$（n，字符或数字）

得到一个有n个字符组成的串。例如：STRING$（30，"*"）得到有30个星号的串。STRING$在制图、制表等方面是非常有用的。
```
5   CLEAR 100
10  PRINT STRING$ (30, "*")
20  PRINT STRING$ (30, 65)
      (65是A的ASCⅡ码)
```

显示结果：

```
＊＊＊…………＊＊＊}
                  } 共30个。
AAA…………AAA     }
```

14. VAL（串）

执行STR 函数的反转换。即把串变量中的字符转换成相应的数字。例如：

```
10    A$ ="12"：B$ ="34"
20    A=VAL (A$ +"E"+B$)；PRINTA，B
```

则有：　A = 12.34，B = 12E34 = 12×10^{34}

如果串的前面是数字，后面是字符，则略掉后面的字符部分。比如：

VAL（"100DOLLARS"）得到100

以上我们介绍了字符串的若干操作，下面给出一例：

例如：INSTRING 子程序。该程序可以判定输入的两个字符串是否是一个包含着另一个。

```
5     CLEAR1000：CLS
10    INPUT "输入长字符串"；X$
20    INPUT "输入短字符串"；Y$
30    GOSUB  1000
40    IFI=0  THEN  70
50    PRINT  Y$；"␣是␣"；X$；"␣的子串"。
55    PRINT  "起始位置"，"结束位置"
60    PRINT  I，I+LEN (Y$) -1
65    PRINT  :PRINT :GOTO 10
70    PRINT  Y$"␣不包含在"；X$；"␣中"
```

```
80      GOTO  10
1000    FOR  I=1 TO LEN(X$)-LEN(Y$)+1
1010    IF  Y$=MID$ (X$,I,LEN(Y$))
                RETURN
1020   NEXT : I=0:RETURN
```

§5.5 数学函数

本节给出的数学函数，若未加说明，自变量一律为单精度常量、变量或表达式，三角函数自变量一律以弧度表示。

1. ABS(X)

   取自变量的绝对值。

2. ATN(X)

   取自变量的反正切（以弧度表示）

3. CDBL(X)

   取自变量的双精度表示。

4. CINT(X)

   取不大于自变量的最大整数。例如CINT(1.5)得1；CINT(-1.5)得-2。自变量X必须在-32768～+32767之间。

5. COS(X)

   取自变量的余弦（自变量必须为弧度）

6. CSNG(X)

   取自变量的单精度数。

7. EXP(X)

   取X的自然指数，即e^x。

8. FIX(X)

   取自变量的整数部分。即把自变量尾数截掉后的整数。

例如：FIX(2.2)＝2，FIX(－2.2)＝－2

9. INT(X)

取自变量的整数部分，得到不大于自变量的最大整数。自变量不受－32768～＋32767的限制。

10. LOG(X)

取自变量的自然对数，即 $\log e^{(x)}$。

11. RANDOM

该函数打开一个随机数发生器。如果程序中使用RND语句就需要把RANDOM放在程序头上。这保证一开机就得到一个不可予测的随机数序列。

12. RND(X)

利用已有的伪随机数（内部产生，不受用戶干涉，产生另一个随机数，并且RND(0)得到一个0～1之间的单精度值。RND（整数）得到一个1～该整数之间的整随机数。但自变量的值必须大于等于零，小于32768。

13. SGN(X)

“符号”函数。X为负数时得－1，X为零时得零；X为正数时得＋1。

14. SIN(X)

取自变量的正弦（自变量必须以弧度表示）

15. SQR(X)

取自变量的平方根。SQR(X)和X [(1/2)是相同的。但算得快。

16. TAN(X)

取自变量的正切（自变量必须以弧度表示）。

§5.6 特殊功能和逻辑运算

1. MOVE(X, Y)

为了在屏幕上作图，把屏幕分成336(水平)×192(垂直)个点。坐标原点在屏幕左上角，由于是从0算起的，所以必须有 $0 \leq X \leq 335$，$0 \leq Y \leq 191$。MOVE(X,Y)表示'抬笔'走到(X,Y)点处。自变量X，Y可以是数字常数、变量或表达式。它们不需要是整数，因为计算机会自动取整数部分。

2. SET(X,Y)

由"笔的当前"位置到(X，Y)处画一条直线。X，Y的限制与MOVE中的相同。

3. RESET(X,Y)

把"笔的当前"位置到(X，Y)处的直线上的所有点变暗。即清一条直线。如果不使用MOVE (X，Y)，上次的SET (X,Y) 或RESET (X,Y) 的终点变成笔的当前位置。

例如：在屏幕中间闪烁一个正方形，可用下面程序。

```
10  CLS: MOVE (100,100)
20  SET (120,100) : SET (120,120)
30  SET (100,120) : SET (100,100)
40  RESET(120,100): RESET (120,120)
50  RESET(100,120): RESET (100,100)
70  GOTO 20
```

4. ERL

取产生错误的程序语句标号。这个功能主要用于错误处理程序中。错误处理程序由ON ERROR GOTO转入。出错语句的标号放在ERL中。

如果错误以直接的方式产生，就得到65535（由2个字节表达的最大数）。应用ERL的程序举例：

```
5     ON ERROR GOTO 1000
10    ……
20    ……
……
1000  PRINT ERL
1010  ……
……
```

5. ERR/2+1

功能与ERL相近，只是EBR中得到的是与错误代码有关的值而不是出错语句标号，通常用于由ON ERROR GOTO语句规定的错误处理程序中ERR/2+1=眞正的错误代码(眞正的错误代码-1）*2=ERR错误代码参见附录5。

6. INP(入口)

从指定的入口取一个字节的数值，例如：

```
10  PRINT  INP(50)
```

意思是：从入口50输入一个字节并打印这个字节的十进制值。总共允许256个口，但H-01机系统已经占用了一部分口。用戶不可使用。参阅附录8。

7. MEM

求内存中尙未使用的字节数。该功能可用于程序中或命令方式中输入PRINT MEM↲得到还有多少空间可为用戶所用。例如：

```
100  IF MEM<80 THEN 900
110  DIM A(15)
```

在分配数组前，用MEM检查一下空间內存大小，以免出现內存溢出（OM）错误。MEM不需自变量。

8. OUT 出口，值

向指定的出口输出一个字节的数值。例如：

```
10  OUT  10,255
```

表示将255送为10端口。两个自变量都限于0～255，但如在INP中所说，系统已经占了的一些口地址，用戶不能使用。

9. POKE 地址，值

将一个0～255之间的值输送到指定內存地址中去。若POKE（或PEEK）的地址超过32767时，应用下面公式：

－1×（65536－所需地址）＝POKE 或 PEEK 地址。

由于POKE语句可将信息存到內存任何地方。因此，一方面它能给使用带来方便和灵活性；另一方面，如果POKE语句破坏了某些重要的系统参数的话，后果难以预料，将不得不从初始化重新开始。

11. POS(X)

该函数获得一个0到41之间的数。表示光标正处在显示器上的位置。POS需要一个“虚自变量”（任何数学表达式），例如：

```
100  PRINT  TAB (20) POS (0)
```

在位置20打印数字20（由于“2”前面插入一个空格放正负号，因此，“2”的实际位置在位置21）。

12. USR(X)

调用一个机器语言子程序。并将自变量传送给子程序。自变量必须是－32768～＋32767之间的一个整数。在不使用自变量时，X 是一个虚变量，例如：

```
100  ……
110  ……
120  X=USR(X)
130  ……
```

第 120 句即将调用一个机器语言子程序。子程序的起始地址应事先用POKE语句存入16910和16911两单元中。16910存低位，16911 存高位。如果要将自变量代入子程序。子程序的第一条语句应是 CALL 1610H，该子程序将取来的自变量放在 HL 寄存器中。

有两种由机器语言子程序返回 BASIC 的方法。

1）如果是不带值返回，可直接在子程序中使用RET.

2）当带值返回时，要把 2 个字节的有符号整数送入 HL 寄存器中，执行一条 JP 162 BH 指令。HL 内容作为有符号数返回。

例如：汇编子程序由 7DOOH 开始，你可用下列程序调用：

```
……
……
100  POKE 16910, 0；入口地址低八位
110  POKE 16911,125；125=7DH 入口地址
                         高八位
120  X=USR (N)
130  ……
```

由于入口地址可以用POKE语句修改，实际上在你的程序中可以调用多个机器语言子程序。

USR 函数被分配有 16 个字节的堆栈区。如果你需要更多的堆栈空间，你可以将 BASIC 的栈指针保存起来，重新设栈区。

要建立用USR函数调用机器语言子程序，你必须在内存高端保留一块内存区。在开机后提示的第一帧画面上，有一行提示“造字起始地址？”，输入一个键后又提示“内存最高地址？”。如果此时你只输入回车键，BASIC 解释程序将使用全部内存空间。如果你让 BASIC 解释程序给你留一块内存，在“内存最高地址？”处键入一个最高地址值，并且有：

最高地址值＝造字起始地址－你想保留的字节数

13. VARPTR（变量名）

求变量的内存地址。如果你指定的变量名没有赋值，使用该函数时会产生 FC（非法调用函数）错误。

如果VARPTR（整变量）得到地址 K，一个整变量占二字节。

地址 K　　包含整数（补码形式）的最低有效数位（LSB）。

地址K＋1　　包含整数的最高有效位（MSB）

如果VARPTR（单精度变量）得到地址 K，一个变量占四个字节。

（K）＝值的 LSB

（K＋1）＝次最高有效位（NEXT MSB）

（K＋2）＝最高有效位（MSB）

（K＋3）＝值的指数

如果 VARPTR（双精度变量）得到 K

(K) =值的 LSB

(K+1)=NEXT MSB

(K+…)=NEXT MSB

(K+6)=MSB

(K+7)=指数

如果 VARPTR=（串变量）得到 K

(K) =串的长度

(K+1)=串起始地址的 LSB

(K+2)=串起始地址的 MSB

对于单精度和双精度变量，数是以规约化的指数形式存贮的。因此，小数点在 MSB 之前，指数已加上了128而且，在 MSB 中的最高一位是用做符号位的。例如：

A! =4　　将存贮如下

4=100(二进制)　　归约化为　1E3

因此，4 的指数为 128+3=131，A的 MSB是100000000 然而由于值是正的，高位为 0。因此，A! 这样存贮。

	指数	MSB	NEXT MSB	LSB
A!=4	131	0	0	0
A!=-5	128	128	0	0
A!=7	131	96	0	0
A!=-7	131	224	0	0

14. 逻辑运算

我们在本章一开始介绍 AND，OR，NOT 怎样用于关系表达式、例如：

```
100 IF A=C AND NOT(B>O)THEN 60 ELSE 50
```

AND、OR和NOT 也用于位（bit）操作，按位比较和进

行逻辑运算，它们首先将其自变量转换成16位的，在-32768～+32767内的带符号补码整数，然后对它们执行指定的逻辑运算，并得到相当范围的结果。如果自变量不在此范围内，产生FC（非法调用）错误。运算是逐位进行的。下面的真值表给出了位之间的逻辑关系：

算符	自变量1	自变量2	结果
AND	1	1	1
	1	0	0
	0	1	0
	0	0	0
OR	1	1	1
	0	1	1
	1	0	1
	0	0	0
NOT	1		0
	0		1

例如：

63AND 16=16　　∵63=111111(2)，16=10000(2)
∴结果为 10000(2)=16

-1AND 8=8　　∵-1=1111111111111111(2)，
8=1000(2)
∴结果为1000(2)=8

4OR 2=6　　∵4=100(2)，2=10(2)
∴结果为 110(2)=6

-1OR -2=-1　　∵-1=1111111111111111(2)，
-2=1111111111111110
∴结果为1111111111111111(2)=-1

NOT 0 = −1	16位 0(2)的反码是16个 1(2),即 −1, ∴NOT −1 = 0
NOT X = −(X + 1)	∵要形成一个数的 16 位补码，则取它的反码并加 1(2)

附录1. H-BASIC的保留词汇

@	CLEAR	CDBL	DEFINT
ABS	CLOSE	CVD	DEFSNG
ASC	CMD	CVS	DEFSTR
AND	CLS	CVI	DEFUSR
ATN	CONT	DATA	DELETE
CHR $	COS	DEFDBL	DIM
CINT	CSNG	DEFFN	EDIT
ELSE	KILL	NOT	SET
END	LEFT $	ON	SGN
EOF	LET	OPEN	SIN
ERL	LSET	OUT	SQR
ERR	LEN	PEEK	STEP
ERROR	LINE	POINT	STOP
EXP	LIST	POKE	STRING $
FIELD	LOAD	POS	STR $
FIX	LOC	PRINT	TAB
FOR	LOF	PUT	TAN
FRE	LOG	RANDOM	THEN
GET	MEM	READ	TIME $
GOSUB	MERGE	REM	TROFF
GOTO	MID $	RFSET	TRON
IF	MKD $	RESTORE	USING

续上表

INKEY $	MKI $	RESUME	USR
INP	MKS $	RETURN	VAL
INPUT	NAME	RIGHT $	VARPTR
INSTR	NEW	RND	COLOR
INT	NEXT	SAVE	MODE
SOUND			

附录2：程序限制(数据范围)

整数	-32768～+32767
单精度数	-1.701411E+38～+1.701411E+38
双精度数	-1.701411834544556E+38～ +1.701411834544556E+38
字符串	最多64个字符
语句标号	0～65529
程序行长度	最多64个字符

附录3：常用内存数量

程序行最少需5个字节

语句标号——2个字节

行指示符——2个字节

回　　车——1个字节

另外，每个保留词、算符、变量名、殊特符号和常数符号都需要1个字节。

附录4：动态(运行时)内存分配

整　变　量：	每个5个字节（数值2个字节，变量名3个字节）
单精度变量：	每个7个字节（数值4个字节，变量名3个字节）
双精度变量：	每个11个字节（数值8个字节，变量名3个字节）
字符串变量：	至少6个字节（变量名3个字节，堆栈和变量指示符3个字，每个字符1个字节）
数组变量：	至少12个字节（变量名3个字节，大小2个字节，维数1个字节，每一维2个字，数组中每一元素2，3，4或8个字节）

每个有效的FOR—NEXT循环：需16个字节

每个有效的（还没有返回）GOSUB：需6个字节

每重括号需4个字节加每个中间值的12个字节

附录5：错误代码

代码	简写	错　　误
1	NF	有NEXT无FOR
2	SN	句法错误
3	RG	有Return无GOSUB
4	OD	数据出界
5	FC	非法调用函数
6	OV	溢出
7	OM	内存出界
8	UL	未定义行
9	BS	数组下标出界
10	DD	重复定义数组（维数）
11	/0	被 0 除
12	ID	非法指令
13	TM	类型不符合
14	OS	字符串空间出界
15	LS	字符串太长
16	ST	字符串公式太复杂
17	CN	不能继续
18	NR	无RFSCME
19	RW	有RESUME无错误
20	UE	不能印出的错误
21	FO	遗漏运算量
22	MD	文件数据不好

错误信息的解释

NF	有NEXT无FOR：使用了NEXT而没有对应的FOR语句。这个错误也可能发生在一个嵌套的循环里保留着NEXT变量的语句。
SN	句法错误：这通常是由于不正确的标点符号，不闭合的括号，错误的字符或者拼写错误的命令等引起的。
RG	有RETURN无GOSUB：在对应的GOSUB被执行以前碰到了RETURN语句。
OD	数据出界：执行READ或INPUT#语句而可用的数据不够，DATA语句可能已经丢失或者所有数据已经从磁带或DATA读完。
FC	非法调用函数：企图执行用非法参量的运算，例如：负自变量的平方根，负的矩阵维数，LOG自变量是负的或0等等，或者调用USR没有首先POKE进入口地点。
OV	溢出：输入或导出的数值对于计算机来说太大或太小了。
OM	內存出界：所有可用的內存已经用完或保留。这可能产生在很大的矩阵维数，嵌套像GOTO，GOSUB这样的转移和FOR-NEXT循环等情况。
UL	未定义的行：企图涉及或转移到不存在的行。
BS	下标超出范围：企图赋值给下标超出维数定义范围的矩阵元素。

DD	重复定义数组：企图定义一个前边已经用DIM或不用语句① 定义过的数组。将所有DIM语句放在程序开头是个好主意。
/0	被零除：企图用0值作分母。
ID	非法指示：将INPUT作直接命令使用。
TM	类型不符：企图把字符串赋给非字符串变量或者反其道而行之。
OS	串空间出界：超出了分配给的字符串（存储）空间总数。
LS	字符串太长：赋给串变量的字符串值的长度超过了255个字符。
ST	字符串公式太复杂：要处理的串运算太复杂，可分成短的段运算。
CN	不能继续：CONT 用于不能继续的程序处，例如在程序结束或编辑之后。
NR	无RESUME：程序达到错误捕获状态，在处理完错误以后找不到出口。
RW	有 RESUME 无 ERROR：在执行 ON ERROR GOTO 以前碰到了RESUME。
UE	不能印出的错误：企图用 ERROR 语句以无效的代码产生错误。
MO	遗漏运算量：没有提供需要的运算量企图作运算。
FD	不合适的文件数据：从外部源(即磁带)输入的数据是不正确的或是以不合适的顺序，等等。

① 不用语句定义是指自动按规定数组每维10个单元

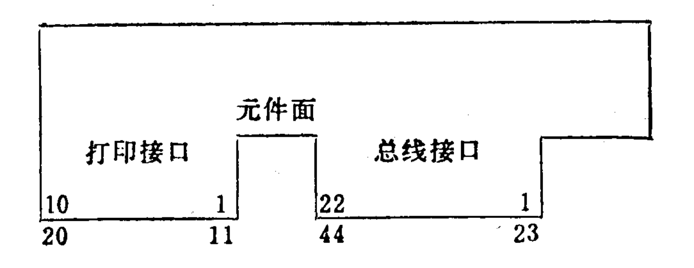
附录6：ASCⅡ码和图示代码

32		33	!	34	″	35	#	36	$
37	%	38	&	39	’	40	(	41	)
42	*	43	+	44	,	45	—	46	.
47	/	48	0	49	1	50	2	51	3
52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=
62	>	63	?	64	@	65	A	66	B
67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L
77	M	78	N	79	O	80	P	81	Q
82	R	83	S	84	T	85	U	86	V
87	W	88	X	89	Y	90	Z	91	[
92	\	93	]	94	^	95	_	96	@
97	a	98	b	99	c	100	d	101	e
102	f	103	g	104	h	105	i	106	j
107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t
117	u	118	v	119	w	120	x	121	y
122	z	123	{	124		125	}	126	
127	Ç	128	ü	129	é	130	â	131	ä
132	à	133	å	134	ç	135	ê	136	ë
137	è	138	ï	139	î	140	ì	141	Ä
142	Å	143	É	144	æ	145	Æ	146	ô
147	ö	148	ò	149	ô	150	ù	151	”

续上页表

152 Ö	153 Ü	154 ç	155 £	156 ¥
157 ₧	158 ƒ	159 á	160 í	161 ó
162 ú	163 ñ	164 Ń	165 ª	166 º
167 i	168 ½	169 ¼	170 i	171 ≪
172 ≫	173 ā	174 ǎ	175 ō	176 ǒ
177 ē	178 ě	179 ī	180 ǐ	181 ū
182 ǔ	183 ü	184 ǘ	185 ǚ	186 ù
187 ∝	188 β	189 Γ	190 π	191 Σ
192 σ	193 μ	194 τ	195 Φ	196 θ
197 Ω	198 δ	199 ∞	200 ϕ	201 ∈
202 ∩	203 ≡	204 ±	205 ⩾	206 ⩽
207 ∫	208 ÷	209 ≈	210 ✓	211 ⁿ
212 ²	213 ∉	214 ⊂	215 ⊇	216 ⊆
217 ⊇	218 {	219 {	220	221

附录7：外引脚定义



打印机接口引线表

表 1

引脚号	信号 名 称	引脚序	信 号 名 称
1	+5V	11	$\overline{IORQ}$
2	A_1	12	D_3
3	D_2	13	D_1
4	D_0	14	$\overline{RD}$
5	A_0	15	A_5
6	A_6	16	D_6
7	D	17	A_4
8	$\overline{WR}$	18	D_6
9	D	19	A_7
10	地	20	地

总线接口引线表

表 2

引脚号	信 号 说 明	引脚号	信 号 说 明
1	地	23	地
2	$\overline{RESET}$	24	A_{11}
3	A_{10}	25	A_{12}
4	A_9	26	A_{13}
5	A_8	27	A_{14}
6	A_7	28	A_{15}
7	A_6	29	$\overline{CLK}$
8	A_5	30	D_4
9	A_4	31	D_3
10	A_3	32	D_5
11	A_2	33	D_6
12	A_1	34	$\overline{8MC}$
13	D_2	35	A_0
14	D_7	36	D_0
15	$\overline{RFSH}$	37	D_1

续表

引脚号	信号说明	引脚号	信号说明
16	$\overline{MI}$	38	$\overline{INT}$
17	$\overline{WAIT}$	39	$\overline{HALT}$
18	$\overline{NHI}$	40	$\overline{MREQ}$
19	$\overline{RD}$	41	$\overline{WR}$
20	$\overline{IORQ}$	42	$\overline{NC}$
21	+5V	43	$u_{32}\ \overline{cs}$
22	地	44	地

附录8：监控主要子程序调用

调用格式：LD　A，子程序调用号
　　　　　RST　30H

子程序调用号	功　　　能
A＝1	从键盘上接收一个字符。字符码值放在BC中。 当为{汉字　B＝国标高位，C＝国标低位 　　　英文　B＝23H，C＝ASCⅡ值
A＝4	打印一个字符，并修改打印位置。BC同上。
A＝5	显示一个字符，并修改光标位置。BC同上 显示控制代码： 08H：退路，并删除该字符 0DH：换行/回车。 0EH：开光标 0FH：关光标 18H：退格，光标← 19H：进格，光标→ 1AH：向下，换行↓ 1BH：向上，换行← 1CH：光标回原点(0,0) 1DH．光标移到行的开头。

子程序调用号	功　　　能
A = 5	1EH　从光标位置起删除行的末尾． 1FH：从光标位置起清除帧的末尾．

附录9：可用的I/O口

Z80CPU共有256个I/O端口，除去下面本系统占用的外，其余可用由用戶随意使用。系统占用的口有：

I/O口	功　　能
00H—F0H	打印机
50H—5FH	读磁带端口
60H—6FH	6845 CRTC 端口
70H—7FH	喇叭，存储器体切换端口 录音机写带。

如果不插打印机接口，00H—0FH号I/O端口也可由用戶使用。

附录10：特殊功能键及键盘命令表

状态	特殊功能键或命令格式	功能	举例及备注
命令状态	RETURN	把刚打入的程序行存入内存；对无标号的语句，命令解释并执行使光标换行。	即回车键，有的计算机用 ENTER 表示，本书用符号↲代表。
	←	光标退回一格并消除光标前一个字符。	
	SHIFT →	将显存内容用打印机拷贝下来。	即屏幕硬拷贝，将屏幕上的内容拷贝到打印机去。
	CLEAR[n]	将所有变量置零并分配n个单元给字符串变量。省略n时仅将变量置零。	

命令状态	AUTO[mm[,nn]]↵	从mm行开始自动编标号，增量为nn。mm和nn为10时可省略	AUTO ↵ AUTO5,5 ↵ AUTO100 ↵ 退出AUTO用 ESC 键
	:	同一程序行中两语句之间的分隔符。	100 A=B:C=20:1 =1
	LIST[mm-nn]↵	将mm到nn行的程序显示出来省略参量时显示整个程序内容。	LIST LIST 20—40 ↵ LST 50— ↵（显示50句以后） LIST—90 ↵（显示90句以前）
	LLIST[mm-nn]↵	同上，只是把程序清单输出到打印机去。	
	DELETE[mm[-nn]]↵	把程序中mm行到nn行删去。	DELETE 100↵（删100句这一行 DELETE20—70↵

续上表

状 态	特殊功能键或命令格式	功　　能	举 例 及 备 注
命令状态	EDIT mm ↵	使标号为mm这一行进入删改状态。	EDIT 90
	RUN[mm]	从标号为mm这一行开始运行程序，省略mm时，从头开始执行。	RUN ↵
	CONT ↵	把用 ESC 或STOP语句暂停的程序继续从断点开始运行。	若暂停期间对程序作了删改，则不能再用CONT命令恢复运行。
	TRON ↵	对正在运行的程序进行跟踪	在打RUN命令前打入该命令。参见TRON语句介绍。
	TROFF ↵	退出跟踪命令。	详细见语句介绍

命令状态	SYSTEM↵	为从录音带上装入机器语言文件而进入监控状态。	
	CSAVE “文件名” ↵	将内存中的程序录制到磁带上。	CSAVE “K” ↵ 具体见磁带机操作一章
	CLOAD “文件名” ↵	把磁带上的BASIC程序装入内存中，并首先清除内存。	具体见磁带机操作一章
	CLOAD?“文件名” ↵	把刚录在磁带上的程序与内存中的程序进行比较，有错则显示“坏”需重录。	详见磁带机操作一章
执行状态	@	暂停程序的运行，或在LIST期间冻结显示，按任意键可恢复运行。	
	ESC	中断程序运行，打CONT命令可继续执行；退出AUTO语句。	即BREAK键，中断。

续上表

状 态	特殊功能键或命令格式	功　　能	举 例 及 备 注
执行状态	RETURN	解释、执行用INPUT语句从键盘输入的数据。	即回车键，本书用"↲"符号代表。
删改状态	[n]空格键	使光标向右移动n格，省略n时只移一格。	
	RETURN	结束并存入已作出的删改，返回命令状态。	
	SHIFT－∧	退出删改子命令X，I，H，但仍处于删改状态。	按下 SHIFT 的同时按下 ∧
	nS(要寻找的字符)	从光标当前位置起找到第n次出现的字符，并把光标移至该处。	例：3SY 把光标移到第3次出现Y处

删改状态	X	光标移至程序行尾，并可增加字符或用 [←]	删改子命令，用 [SHIFT－∧] 退出。
	H	从光标当前位置起删除程序行中未显示部分，并可增加字符或用 [←] 删去字符。	删改子命令，用 [SHIFT －∧] 退出
	I	从光标当前位置起插入所需内容，或用 [←] 删去字符。	同上。
	[n] D	从光标当前位置起，向右删去 n 个字符。省略 n 时只删去一个。	4D：删去 4 个字符 D：删去一个字符
	[n] C(n个新字符)	从光标当前位置起，用C 后面打入的 n 个新字符代替原来的n个字符，省略n时只替换1个	3CB：用BBB代替原来的 3 个字符 CA：用A代替原来的一个序符

续上表

状 态	特殊功能键或命令格式	功　　能	举例及备注
删	[n]K(x)	删去从光标当前位置起至第n次出现的字符x间的所有字符。	2KM：删去从光标当前位置到第2次出现的M之间的所有字符
改	A	撤消已作过的删改并仍处于删改状态。	
状	Q	撤消已作过的删改，返回到命令状态。	
态	E	保存已作过的删改，返回到命令状态。	